

CONTINUING (1.53(b)) UTILITY PATENT APPLICATION TRANSMITTAL

(Only for continuing applications under 37 CFR 1.53(b))

Attorney Docket No.
1364.1001D5/RAG

First Named Inventor: Dale T. ROBERTS

Title: System for Collecting Use Data Related to Playback of Recordings

Express Mail Label No.

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: **Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231**

1. ☒ Fee Transmittal Form
2. ☒ Specification, Claims & Abstract [Total Pages: 27]
3. ☒ Drawing(s) (35 USC 113) [Total Sheets: 3]
4. ☒ Oath or Declaration [Total Pages: 1]
 - a. ☐ Newly executed (original or copy)
 - b. ☒ Copy from a prior application (37 CFR 1.63(d)) (see Box 18)
5. ☐ This application is filed by fewer than all the inventors named in the prior nonprovisional application.
 - a. ☐ **DELETE** the following inventor(s) named in the prior nonprovisional application:
 - b. ☐ The inventor(s) to be deleted are set forth on a separate sheet attached hereto.
6. ☒ Incorporation by Reference (usable if Box 4b is checked)
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
7. ☐ Microfiche Computer Program (Appendix)
8. ☐ Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
 - a. ☐ Computer Readable Copy
 - b. ☐ Paper Copy (identical to computer copy)
 - c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. ☐ Assignment Papers (cover sheet & document(s))
10. ☐ 37 CFR 3.73(b) Statement (when there is an assignee) [☐ Power of Attorney
11. ☐ English Translation Document (if applicable)
12. ☐ Foreign priority benefit under 35 U.S.C. §119 is claimed.
 - a. ☐ Certified Copy of Priority Document(s) filed in prior application No. / .
 - b. ☐ Certified Copy of Priority Document(s) enclosed.
 - c. ☐ Certified Copy of Priority Document(s) to follow.
13. ☒ Information Disclosure Statement (IDS)/PTO-1449 [☐ Copies of IDS Citations
14. ☒ Preliminary Amendment
 - a. ☒ enclosed herewith.
 - b. ☐ incorporated herein (see Box 18).
15. ☒ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
16. ☒ Small Entity Statement(s) [☐ Statement filed in prior application, status still proper and desired.
17. ☐ Other:

18. CONTINUING APPLICATION, check appropriate box and supply the requisite information below:

☒ Continuation ☒ Divisional ☐ Continuation-in-part (CIP) of prior application No: 09/060,876

Prior application information: Examiner: _____

Group/Art Unit: _____

* * *

Preliminary Amendment:

☐ Cancel in this application original claims _____ of the prior application before calculating the filing fee.
(At least one original independent claim must be retained for filing purposes.)

☐ Amend the specification by inserting before the first line the sentence:

--This application is a _____ of application number _____, filed _____, now _____.--

19. NEW CORRESPONDENCE ADDRESS * CUSTOMER NO. 21,171

STAAS & HALSEY LLP
700 Eleventh Street, N.W.
Suite 500
Washington, DC 20001

Telephone: (202) 434-1500
Facsimile: (202) 434-1501

20. SIGNATURE OF ATTORNEY OR AGENT

NAME	Richard A. Gollhofer	REGISTRATION NO.	31,106
SIGNATURE	<i>Richard A. Gollhofer</i>	DATE	8/24/99

Applicant or Patentee: Dale Tyson ROBERTS et al.Serial or Patent No.: (Div. of 09/060,876)Filed or Issued: (Concurrently)For: SYSTEM FOR COLLECTING USE DATA RELATED TO PLAYBACK OF RECORDINGS

**VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY
STATUS (37 CFR 1.9(f) and 1.27(c)) -- SMALL BUSINESS CONCERN**

I hereby declare that we are

- ☐ the owner of the small business concern identified below:
☒ officials of the small business concern empowered to act on behalf of the concern identified below:

NAME OF CONCERN CDDB, INC.ADDRESS OF CONCERN 12955 Old Meridian Street, Suite 107Carmel, IN 46032

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.3-18, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third-party or parties controls or has the power to control both.

I hereby declare that all rights under contract or law have been conveyed, to and remain with the small business concern identified above with regard to the invention, entitled SYSTEM FOR COLLECTING USE DATA RELATED TO PLAYBACK OF RECORDINGS

by inventor(s) Dale Tyson Roberts et al.

described in

- ☒ the specification filed herewith.
☐ application serial no. _____, filed _____.
☐ patent no. _____, issued _____.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights to the invention is listed below* and no rights to the invention are held by any person, other than the inventor, who could not qualify as a small business concern under 37 CFR 1.9(d) or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

***NOTE:** Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities (37 CFR 1.27).

NAME

ADDRESS

☐ INDIVIDUAL ☐ SMALL BUSINESS CONCERN ☐ NONPROFIT ORGANIZATION

NAME

ADDRESS

☐ INDIVIDUAL ☐ SMALL BUSINESS CONCERN ☐ NONPROFIT ORGANIZATION

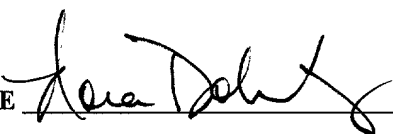
I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small business entity is no longer appropriate. (37 CFR 1.28(b)).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING Nora Doherty

TITLE OF PERSON OTHER THAN OWNER Vice President

ADDRESS OF PERSON SIGNING 12955 Old Meridian Street, Suite 107, Carmel, IN 46032

SIGNATURE 

Date: Aug 18, 1999

S&H 8/99

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Dale Tyson ROBERTS et al.

Serial No.: (Div. of 09/060,876)

Group Art Unit:

Filed: (Concurrently)

Examiner:

For: SYSTEM FOR COLLECTING USE DATA RELATED TO PLAYBACK OF
RECORDINGS (As Amended)

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Before examination of the above-identified application, please amend the application as follows:

IN THE TITLE:

Replace the title with --SYSTEM FOR COLLECTING USE DATA RELATED TO
PLAYBACK OF RECORDINGS--.

IN THE SPECIFICATION:

Between the title and the heading "BACKGROUND", insert a new section entitled:

--CROSS-REFERENCE TO RELATED APPLICATIONS

This is a divisional of co-pending U.S. Patent Application Serial Number 09/060,876
filed April 15, 1998, now pending.--

IN THE CLAIMS:

Please CANCEL claim 1 and ADD the following new claims:

2. A method of transferring information associated with playback of a recording at a first device connected to a second device via a network, comprising:

collecting use data associated with the recording; and

transferring the use data from the first device to the second device via the network.

3. A method as recited in claim 2, wherein the use data are related to at least one of a portion of the recording and play time of the recording.

4. A method as recited in claim 3, further comprising:

obtaining user demographic data; and

sending the user demographic data to the second device.

5. A method as recited in claim 4, further comprising supplying complementary content for the recording to the first device via the network.

6. A method as recited in claim 5, further comprising selecting the complementary content based on the use data and the user demographic data.

7. A method as recited in claim 2,

wherein the local device is a computer, and

wherein said collecting includes monitoring operations performed by the computer while the recording is played.

8. A method as recited in claim 7,

wherein said monitoring detects software executing on the computer while the recording is played, and

wherein the use data transferred to the second device include identification of the software executing on the computer while the recording is played.

9. A method as recited in claim 7, wherein said monitoring detects user inputs while the recording is played.

10. A method as recited in claim 7, wherein said monitoring detects operational data transmitted over the network to and from the computer.

11. A method as recited in claim 10, wherein the operational data include at least one uniform resource locator.

12. A method as recited in claim 11, wherein said method further comprises supplying complementary content for the recording to the first device via the network.

13. A method as recited in claim 12,
wherein the use data include the operational data, and
wherein said method further comprises:
 sending user demographic data from the first device to the second device
via the network; and
 selecting the complementary content based on the use data and the user
demographic data.

14. A method as recited in claim 7,
wherein said collecting detects software loaded in the computer, and
wherein the use data transferred to the second device include identification of
the software loaded into the computer.

15. A method as recited in claim 14,
wherein said collecting detects frequency of use of the software loaded in the
computer, and

wherein the use data transferred to the second device include the frequency of use of the software loaded in the computer.

16. A method as recited in claim 2,
further comprising registering a user of the first device to obtain a user identifier, and
wherein said transferring further includes transferring the user identifier with the use data to the second device.

17. A method as recited in claim 16, further comprising:
obtaining user demographic data; and
sending the user demographic data to the second device.

18. A method as recited in claim 17, further comprising supplying complementary content for the recording to the first device via the network.

19. A method as recited in claim 18, further comprising selecting the complementary content based on the use data and the user demographic data.

20. A method as recited in claim 2, wherein the use data include frequency data indicating number of times the recording is played at the first device.

21. A method as recited in claim 20, wherein the recording contains segments, and
wherein the frequency data indicate the number of times the segments are played at the first device.

22. A method as recited in claim 2, wherein the recording contains segments,
wherein said method further comprises identifying at least one segment of the recording, and

wherein the use data include segment data identifying the at least one segment played at the first device.

23. A method as recited in claim 22, wherein the segment data include a number of times the at least one segment has been played at the first device.

24. A method as recited in claim 23, wherein the segment data include how much of the at least one segment has been played each time.

25. A method as recited in claim 22, wherein the segment data include how much of the at least one segment has been played.

26. A method as recited in claim 22, wherein the segment data include how often the at least one segment has been played.

27. A method as recited in claim 22, wherein the recording is a compact disc with audio content, the segments are tracks on the compact disc and the use data indicate playback of the tracks on the compact disc.

28. A method as recited in claim 22, wherein the recording is a digital versatile disc, the segments are tracks on the digital versatile disc and the use data indicate playback of the tracks on the digital versatile disc.

29. A method as recited in claim 22,
further comprising identifying the recording based on length of the segments to obtain a recording identifier, and
wherein said transferring includes sending the recording identifier to the second device.

30. A method as recited in claim 29, wherein the recording is a disc and the length of the segments is obtained from table of contents information on the disc.

31. A method as recited in claim 2, wherein the recording is a computer file containing digitized audio signals.

32. A method as recited in claim 31, wherein the computer file contains audio data compressed using MPEG encoding.

33. A method as recited in claim 2, wherein the use data include total time of use of the recording.

34. A method as recited in claim 33, wherein the use data include time of use of the first device.

35. A method as recited in claim 2, further comprising supplying complementary content for the recording to the first device via the network.

36. A method as recited in claim 35, further comprising:
obtaining user demographic data; and
sending the user demographic data to the second device.

37. A method as recited in claim 36, further comprising selecting the complementary content based on the use data and the user demographic data.

38. A method as recited in claim 2, further comprising continuing said collecting and transferring of the use data from the first device to the second device via the network as long as the recording is being played at the first device.

39. A method as recited in claim 2,
further comprising detecting a volume level at which the recording is being
played, and
wherein said sending includes sending data indicating the volume level.

40. A method as recited in claim 2,
further comprising reading a recording identifier on the recording, and
wherein said transferring includes sending the recording identifier to the second
device.

41. A system for obtaining information based on playback of at least one recording at a
first device connected to a second device via a network, comprising:
collecting means for collecting use data related to at least one of a portion of the
recording and play time of the recording;
sending means for sending the use data from the first device to the second
device via the network; and
storing means for storing the use data at the second device.

42. A system as recited in claim 41,
wherein the use data relate to a plurality of recordings, and
wherein the use data include frequency of use of the recordings.

43. A computer-readable storage storing instructions to control a processor to perform
a process comprising:
collecting use data associated with playback of a recording at a first device; and
transferring the use data from the first device to a second device connected to
the first device via a network.

44. A computer-readable storage as recited in claim 43, wherein said process further comprises:

obtaining user demographic data; and
sending the user demographic data to the second device.

45. A computer-readable storage as recited in claim 44, wherein said process further comprises obtaining as the use data, information regarding playing of at least one portion of the recording.

46. A computer-readable storage as recited in claim 45, wherein said process further comprises supplying complementary content for the recording to the first device via the network.

47. A computer-readable storage as recited in claim 46, wherein said process further comprises selecting the complementary content based on the use data and the user demographic data.

48. A computer-readable storage as recited in claim 43,
wherein said process further comprises monitoring operations performed by the first device during playback of the recording, and
wherein the use data indicate the operations performed by the first device during playback of the recording.

49. A computer-readable storage as recited in claim 48, wherein the operational data include at least one uniform resource locator.

50. A computer-readable storage as recited in claim 49, wherein said process further comprises supplying complementary content for the recording to the first device via the network.

51. A computer-readable storage as recited in claim 50, wherein said process further comprises:

sending user demographic data from the first device to the second device via the network; and

5 selecting the complementary content based on the use data and the user demographic data.

52. A computer-readable storage as recited in claim 43,

wherein the first device is a computer,

wherein said collecting detects software loaded in the computer, and

wherein the use data transferred to the second device include identification of the software loaded into the computer.

53. A computer-readable storage as recited in claim 52,

wherein said collecting detects frequency of use of the software loaded in the computer, and

wherein the use data transferred to the second device include the frequency of use of the software loaded in the computer.

54. A computer-readable storage as recited in claim 43,

wherein said process further comprises registering a user of the first device to obtain a user identifier, and

wherein said transferring further includes transferring the user identifier with the use data to the second device.

55. A computer-readable storage as recited in claim 54, wherein said process further comprises:

obtaining user demographic data; and

sending the user demographic data to the second device.

56. A computer-readable storage as recited in claim 55, wherein said process further comprises supplying complementary content for the recording to the first device via the network based on the use data and the user demographic data.

57. A computer-readable storage as recited in claim 43, wherein the recording contains segments,

wherein said process further comprises identifying at least one segment of the recording, and

wherein the use data include segment data identifying the at least one segment played at the first device.

58. A computer-readable storage as recited in claim 57, wherein the segment data include a number of times the at least one segment has been played at the first device.

59. A computer-readable storage as recited in claim 58, wherein the segment data include how much of the at least one segment has been played each time.

60. A computer-readable storage as recited in claim 57, wherein the segment data include how much of the at least one segment has been played.

61. A computer-readable storage as recited in claim 57, wherein the segment data include how often the at least one segment has been played.

62. A system for transferring information associated with playback of a recording, comprising:

a first device to reproduce a recording and collect use data associated with the recording;

a second device to store the use data; and

a network coupled to said first and second devices to transfer the use data from said first device to said second device.

63. A system as recited in claim 62, wherein the recording is a compact disc with audio content stored on tracks and the use data indicate playback of the tracks on the compact disc.

64. A system as recited in claim 63,
wherein said first device obtains a recording identifier based on length of the tracks, and
wherein said network sends the recording identifier to said second device.

65. A system as recited in claim 62, wherein the recording is a digital versatile disc and the use data indicate playback of tracks on the digital versatile disc.

66. A system as recited in claim 62, wherein the recording is a computer file containing digitized audio signals.

67. A system as recited in claim 66, wherein the computer file contains audio data compressed using MPEG encoding.

68. A system as recited in claim 62, wherein said network transfers complementary content for the recording from said second device to said first device.

69. A system as recited in claim 68,
wherein said first device obtains user demographic data; and
wherein said network sends the user demographic data from said first device to said second device.

70. A system as recited in claim 69, further comprising wherein said second device selects the complementary content based on the use data and the user demographic data.

REMARKS

This Preliminary Amendment is submitted to present claims directed to subject matter disclosed, but not claimed in the parent application.

It is respectfully requested that this Preliminary Amendment be entered in the above-referenced application and that the claims receive favorable examination at the earliest possible date.

If any further fees are required in connection with the filing of this Preliminary Amendment, please charge same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

By: Richard A. Gollhofer
Richard A. Gollhofer
Registration No. 31,106

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
(202) 434-1500

Date: 8/24/99

NETWORK DELIVERY OF
INTERACTIVE ENTERTAINMENT
COMPLEMENTING AUDIO RECORDINGS

BACKGROUND

5 **1. Field of the Invention**

 This invention pertains to the field of computer networking, and more particularly to the use of network protocols to provide services to users which are related to audio recordings.

10 **2. Related Art**

 Over the past few years, on-line services have experienced explosive growth and have become a major new form of entertainment. Alongside this new entertainment, more traditional forms such as musical recordings have continued to be consumed on a massive scale.

15 The traditional experience of the musical recording is listening by a small group of persons gathered together in a room. The music fills the room acoustically, but there is little associated visual content, and there is only a limited interaction with the recording, consisting essentially of deciding which tracks to play and performing simple transformations on the recorded sound, such as setting the volume or applying an audio equalizer. This traditional experience dates back to the early age of 78 r.p.m. musical
20 recordings almost a century ago.

The traditional production of a musical recording complements the traditional experience of the recording. The recording is produced in a number of recording sessions, subject to careful mixing and editing, and then released to the public. At that point, the recording is in a fixed form, nowadays an audio CD, whose purpose is to record as faithfully as possible the final sonic experience designed by its authors, the musicians, producer, and recording engineers.

Music videos have supplemented the traditional experience of musical recordings by allowing the association of visual content with tracks of such a recording. In practice, however, music videos have been broadcast, with all the problems of lack of user control which that implies, and they have not contributed to interactivity or participation by the consumer.

On-line services offer opportunities for enriching the experience associated with musical recordings. The present invention is addressed to computer programs, systems, and protocols which can fulfil this promise.

SUMMARY OF THE INVENTION

It is therefore an object of this invention to provide computer programs, systems, and protocols which allow producers to deliver entertainment complementary to musical recordings by means of on-line services such as the Internet. It is a further object of this invention to provide computer programs, systems, and protocols which allow such complementary entertainment to be meaningfully interactive for the consumer, such that the consumer can also be a creator of the experience.

It is a further object of the invention to achieve the foregoing objects by means of implementations designed to attain integration with existing environments and programs, particularly on the Internet, while retaining the flexibility to adapt to the continuing evolution of standards for on-line services.

In one aspect of the invention, software is provided which permits a computer program running on a remote host to control a compact disk (CD) player, DVD player, or the like on a user's computer. (For convenience, we use the term "CD player" to refer also to DVD players and similar devices.) The software is designed to permit the remote host both to initiate actions on the CD player and to become aware actions which the user has initiated by other control means, such as the buttons on the CD player's front panel or a different CD player control program. This aspect of the invention is a building-block for the provision of complementary entertainment for musical recordings when those recordings are fixed in the prevailing contemporary form, the audio CD.

In a second aspect of the invention, visual content, including interactive content, may be delivered over an on-line service in such a way that it is synchronized to the delivery of content from a musical recording. Such visual content may, for example, be synchronized to the playing of an audio CD in the user's computer. The visual content is thematically linked to the musical recording, for example in the manner of a music video.

In a third aspect of the invention, a method is provided for assigning a unique identifier to musical recordings consisting of a number of tracks. A unique identifier is a useful complement to the delivery of visual content in conjunction with the playing of an audio CD in that it allows the software which delivers the visual content to be sure that the audio CD is in fact the correct CD to which the visual content corresponds. If the visual content is designed, for example, to accompany the *Rosary Sonatas* of Heinrich Ignaz Franz Biber, it would presumably not function well if the CD in the user's player were the soundtrack for the film *Mary Poppins*. The unique identifier also allows a CD to be used as a key to access a premium Web area. Furthermore, the unique identifier can allow the user to be directed to an area of the Web corresponding to the CD which is in the user's machine.

In a fourth aspect of the invention, the immensely popular on-line service generally referred to as a "chat room" may be enhanced by means of a link to a musical recording to which all persons in the room are listening. The chat room experience as it exists today in on-line services has a disembodied quality by comparison with traditional face-to-face social encounters, in which there are identifiable surroundings. The only common experience to the chat users today are the words of the chat as they fly by on a computer screen, and perhaps the user icons ("avatars") or other visual content occupying a small space on the screen. The use of a musical recording in conjunction with a chat room opens up the possibility of restoring to the experience a degree of the shared ambience of traditional social encounters. Furthermore, the musical recording offers a focal point that allows chat-seekers to group together by means of shared interests in a particular type of recording.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of the environment in which the preferred embodiment operates.

Figure 2 is a flowchart of the synchronization code of the invention.

Figure 3 is a flowchart of the sequence of operations for connection to a chat room focused on a musical recording.

DESCRIPTION OF PREFERRED EMBODIMENTS

1. Introduction

The preferred embodiment of this invention operates on the World Wide Web. The software implementation environment provided by the World Wide Web is described in a number of books, for example, John December & Mark Ginsburg, *HTML 3.2 and*

CGI Unleashed (1996). The World Wide Web is based on a network protocol called HTTP (hypertext transfer protocol), which is described in T. Berners-Lee et al., *Hypertext Transfer Protocol — HTTP/1.0* (Internet Request for Comments No. 1945, 1996). The HTTP protocol must be run atop a general connection-oriented protocol, which today is generally TCP/IP, described in Douglas E. Comer, *Internetworking with TCP/IP* (3d ed. 1995). However, the invention described here is not limited to HTTP running over any particular kind of network software or hardware. The principles of the invention apply to other protocols for access to remote information that may come to compete with or supplant HTTP.

As shown in Fig. 1, a Web user sits at his or her computer and runs a computer program called a browser. The browser sends out HTTP requests to other computers, referred to as servers. In requests, particular items of data, referred to as resources, which are available on servers, are referred to by means of uniform resource locators (URL's), character strings in a particular format defined in Berners-Lee et al., *supra*. A URL includes both an identification of the server and an identification of a particular item of data within the server. Reacting to the requests, the servers return responses to the user's browser, and the browser acts upon those responses, generally by displaying some sort of content to the user.

The content portion of the responses can be a "Web page," expressed in the hypertext markup language (HTML). That language allows one to express content consisting of text interspersed with bitmap-format images and links (also known as anchors and hyperlinks). The links are further URL's to which the browser may, at the user's prompting, send further requests.

The responses can also include more complex commands to be interpreted by the browser, e.g., commands which result in an animation. HTML itself does not define complex commands, but rather they are considered to belong to separately-defined scripting languages, of which the two most common ones are JavaScript and VBScript.

In addition to extending the function of the browser by means of code written in a scripting language, it is also possible to extend the function of a browser with compiled code. Such compiled code is referred to as a "plug-in." The precise protocol for writing a plug-in is dependent on the particular browser. Plug-ins for the Microsoft
5 browser are referred to by the name of ActiveX controls.

Plug-ins may be very complex. A plug-in which may advantageously be used in connection with the invention is Shockwave from Macromedia. It permits animations which are part of a server response to be downloaded and played to the user. Shockwave defines its own scripting language called Lingo. Lingo scripts are contained within the downloadable animations which the Shockwave plug-in can play. The general
10 format of a Shockwave animation is a timeline consisting of a series of frames, together with a number of visual objects which appear, perform motions, and disappear at particular frames within the timeline. To achieve more complex effects within a Shockwave animation, Lingo scripts may be invoked in addition to predefined visual
15 objects.

2. Command Plug-In

A preferred embodiment of the invention employs a plug-in, referred to as the command plug-in, which provides to a scripting language the ability to command in a detailed fashion the playing of a musical recording. The command plug-in should
20 provide, at a minimum, the following basic functions:

- (1) *Start and stop play.*
- (2) *Get current track and position within the track.*
- (3) *Seek to a track and a position within the track.*
- (4) *Get and set volume.*
- (5) *Get information regarding the CD (e.g., the number of tracks, their lengths, the pauses between tracks).*

(6) *Get information regarding the capabilities of the CD drive.*

Other functions may be provided, limited only by what the underlying operating system services are able to provide.

5 The command plug-in is preferably written in a conventional programming language such as C++. The plug in must conform to the existing standards for plug-ins, such as those required of Microsoft ActiveX objects. In order to obtain the information and carry out the functions which the command plug-in makes available to the scripting language, the command plug-in relies on functions which provide control and information regarding the playing musical recording. These functions will depend on the precise source of the recording. If, as in the currently preferred embodiment, the recording is being played on an audio CD in the computer CD player, and if the browser is running under Microsoft Windows 3.1 or Windows 95, these functions would be the MCI functions, which form a part of the Win32 application programming interface. These functions are documented, for example, in *Microsoft Win32 Programmer's Reference*. Different functions may be provided by streaming audio receivers, as for example receivers which capture audio which is coming into the user's computer over a network connection in a suitable audio encoding format such as MPEG.

20 An important point to note about the implementation of the command plug-in is that the operations which it carries out, as for example seeks, may take times on the order of a second. It is undesirable for the command-plug in to retain control of the machine during that interval, so it is important that the plug-in relinquish control of the machine to the browser whenever a lengthy operation is undertaken, and report on the results of the operation via the asynchronous event handling capability used in the common scripting languages.

25 Given the above summary of the functions which the command plug-in provides, a general knowledge of how to write plug-ins (e.g., of how to write ActiveX objects),

and a knowledge of the relevant application programming interface for controlling the play of the musical recording (e.g., MCI in Win32), a person skilled in the art could readily and without undue experimentation develop an actual working command plug-in. For this reason, further details of how the command plug-in is implemented are not provided here.

3. Synchronization

The existence of a command plug-in providing the functions listed above to a scripting language is a foundation on which entertainment complementary to a musical recording may be constructed. In particular, it is possible to devise, building on this foundation, a method for synchronizing the display of visual content by means of the scripting language with the events which are occurring on the audio CD.

In a preferred embodiment of the invention, the synchronization of the visual content to the audio CD proceeds as follows. The visual content is provided by means of a Shockwave animation, which is downloaded from the server and displayed for the user by means of a Shockwave plug-in. This downloading may take place before the animation is displayed, or alternatively it may take place as the animation is being displayed, provided the user's connection to the network is fast enough to support download at an appropriate speed. The downloading is a function provided by the Shockwave plug-in itself.

As the Shockwave animation is played, a Lingo script executes each time a frame finishes displaying. The Lingo script contains a description of the relationship which should exist between frames of the animation and segments of the musical recording, identified by track number and by time. The Lingo script determines, by means of the command plug-in described above, at which track and time the play of the audio CD is. It then refers to the description in order to determine which frames of the animation correspond to that portion of the audio CD. If the current frame is not one of those

frames, the Lingo script resets the time line of the animation so that the animation will begin to play at the frame which corresponds to the current position of the audio CD. This permits the visual content to catch up if it ever lags the CD, for example because downloading from the network has fallen behind, because the user's computer lacks the cycles to play the animation at full speed, or because the user has fast forwarded the CD.

In a variant form of this synchronization algorithm (shown in Fig. 2), the frames of the animation are arranged into groups of contiguous frames. A correspondence is established between each such group of frames and a particular segment of the audio recording (box 200 in Fig. 2). At the end of each frame of the animation, the audio play position is determined (box 210). A test is done to determine whether the audio play position is within the segment of the recording that corresponds to the group of frames to which the next sequential frame belongs (box 215). If the audio play position is within that segment, the playback of the animation proceeds with that next frame (box 230). If the audio play position is not within that segment, then the playback of the animation is advanced to the frame corresponding to where the audio is (boxes 220 and 225).

4. Unique Identifiers for Audio CDs

A further aspect of the invention is the ability, by making use of the command plug-in, to provide a technique for establishing a unique identifier for an audio CD which is located in the user's CD player. The unique identifier may be based on the number and lengths of the tracks (measured in blocks, i.e., 1/75ths of a second), so that the identifier would be a concatenation of these lengths. In practice, however, it is desirable to have a somewhat shorter identifier, so the unique identifier is preferably the concatenation of the track lengths expressed in a fairly coarse unit, such as 1/4th of a second.

Appendix A contains source code, written in C, for a fuzzy comparison algorithm suitable for determining whether two audio CDs are exactly or approximately the same. The fuzzy comparison algorithm proceeds as follows. For each of the two audio CDs to be compared, one determines the lengths of all the tracks in the recordings in milliseconds. One then shifts all track lengths to the right by eight bits, in effect performing a truncating division by $2^8 = 256$. One then goes through both of the recordings track by track, accumulating as one proceeds two numbers, the match total and the match error. These numbers are both initialized to zero at the start of the comparison. For each of the tracks, one increments the match total by the shifted length of that track in the first CD to be compared, and one increments the match error by the absolute value of the difference between the shifted lengths of the track in the two CDs. When one gets to the last track in the CD with the fewer number of tracks, one continues with the tracks in the other CD, incrementing both the match total and the match error by the shifted lengths of those tracks. Following these steps of going through the tracks, the algorithm then divides the match error by the match number, subtracts the resulting quotient from 1, and converts the difference to a percentage which is indicative of how well the two CDs match.

Appendix B contains source code, written in C, for a comparison algorithm suitable for determining whether two audio CDs are exactly the same. The algorithm generates from the number of tracks, the track lengths, and the start and end times of the tracks an 8-byte value. The high order 4 bytes are obtained by summing the start and end times of all tracks, expressed in milliseconds. The low order 4 bytes are obtained by summing the lengths of all tracks expressed in milliseconds, shifting the sum left ten bits, and adding the number of tracks.

A unique identifier for a musical recording may be employed as a database key. A site may maintain a database of information about CDs, for example information about all CDs issued by the particular record company can be maintained on that record company's site. There are various alternative ways for users to navigate this

information. For example, they could use a Web page containing many hyperlinks as a table of contents, or they could use a conventional search engine. A third way of searching which is enabled by the unique identifier of the invention is for there to be Web page which invites the user to place in the computer's CD drive the CD about which he or she is seeking information. Upon detection of the presence of the CD in the drive, a script in the Web page computes the unique identifier corresponding to the CD and sends it to the server. The server then displays information about the CD retrieved from a database on the basis of that unique identifier. This information may include a Web address (URL) that is related to the audio CD (e.g., that of the artists' home page), simple data such as the names of the songs, and also complementary entertainment, including potentially photographs (e.g., of the band), artwork, animations, and video clips. It is also possible to arrange things so that, when the user inserts an audio CD into the computer, (i) the Web browser is launched if not already running, (ii) the browser computes the CD's unique identifier and from that unique identifier derives a URL, and (iii) the browser does an HTTP get transaction on that URL.

An alternative application of unique identifiers for musical recordings is to employ an audio CD as a key for entering into a premium area of the Web. There are presently premium areas of the Web to which people are admitted by subscription. A simple form of admission based on the unique identifier is to require, before accessing a particular area of the Web, that the user place in his or her CD drive a particular CD, or a CD published by a particular company or containing the music of a particular band or artist. This is readily accomplished by means of a script which invokes the functions provided by the command plug-in and computes a unique identifier.

5. Chat Rooms Connected With Musical Recordings

A third aspect of the invention is the connection of chat rooms with musical recordings. The goal is to provide all participants in a chat room with the same music at approximately the same time.

The prevailing network protocol for chat services is Internet Relay Chat (IRC), described J. Oikarinen & D. Reed, *Internet Relay Chat Protocol* (Internet Request for Comments No. 1459, 1993). In this protocol, when one becomes a client of a chat server, one sends the name of a chat room. The chat server receives messages from all of its of clients and relays the messages sent in by one client to all the other clients connected in the same room as that client. The messages which a client sends are typically typed in by the user who is running the client, and the messages which a client receives are typically displayed for the user who is running the client to read.

In a preferred embodiment of the invention, a chat client is customized by means of a plug-in, which we will call the chat plug-in. The chat client is started up by a browser as follows (see Fig. 3). The user connects by means of the browser to a central Web page (box 300) which, upon being downloaded, asks that the user insert a CD into his or her player (box 305). A unique identifier of the CD is computed and communicated back to the server by using the control plug-in described above under the command of a script in the central Web page (box 310). The server then employs the unique identifier to determine whether it has a chat room focused on the CD (box 315). This step may be carried out by looking the unique identifier up in a database using techniques well known in the art. There exists a vast literature on connecting Web pages to databases, e.g., December & Ginsburg, *supra*, chapter 21. If a chat room focused on the CD exists or can be created, the server responds with the name of that chat room, and the browser starts up a chat client on the user's computer as a client of that chat room (box 320).

The chat room's name is set by the server to contain information about the track which the CD is playing in the other chat room clients' machines and the time at which the track started to play, as well as about the volume at which the CD is playing. The chat client plug-in employs that information to direct the control plug-in to set the CD in the user's computer to play in such a manner that it is approximately synchronized to the CD which is playing in the other chat room clients' machines (box 320).

Each user in the chat room is able to control the CD which is playing in his or her machine. Control actions result in the chat plug-in sending messages to the chat server which describe the control action being taken (box 325). For example, such messages may indicate a change in the position of the CD, a change in the volume, or the ejection of the CD to replace it with another. The chat plug-ins running on the other users' machines, upon seeing a message of this kind, replicate the action (as far as possible) on the other users' machines by using the control plug-in described above (box 330).

In a further aspect of the invention, a chat room focused on a particular musical recording might allow for a voting procedure to select particular tracks. A simple voting procedure would be for each chat plug-in to act upon a change message of the kind described in the preceding paragraph only when it sees two identical consecutive change messages. This would mean that in order to change the track which is being played, it would be necessary for two users to change to that track. The number two may be replaced by a higher number.

In a further aspect of the invention the messages delivered to the users of a chat can be driven from a text file rather than manual typing. This would allow a pre-recorded experience to be played back for a group of chat users. Such a technique may be used to create a pre-recorded, narrated tour of an audio CD.

An important advantage of the preferred embodiment as described above is that it may be used with any chat server software which supports the minimal functionality required by Internet Relay Chat or by a protocol providing similar minimum chat service. The additional software required is located in the chat client plug-in and in the central Web page, with its connection to a database of CD information.

APPENDIX A

```

5  /*
   * FUZZY CD ID
   * (c) 1996 ION
   *
   * by Ty Roberts
   */

10 #include <stdio.h>
   #include <stdlib.h>
   #include <time.h>

   struct fuzzyCDid {
       short          numTracks; // start time in milliseconds
       unsigned short fuzzlength[100];
15 };

   typedef struct fuzzyCDid fuzzyCDid, *fuzzyCDidPtr;

   // structure of a cd track with all times stored in milliseconds

   struct cdtrack {
       long  beginMs;    // start time in milliseconds
       long  endMs;      // end time in milliseconds
       long  lengthMs;   // length in milliseconds
20 };

   typedef struct cdtrack cdtrack, *cdTrackPtr;

   struct cd {
25     short  numTracks;
       cdtrack track[100];
   };

   typedef struct cd cd, *cdPtr;

   void CreateFuzzyId( fuzzyCDidPtr fid, cdPtr cd );

30 float FuzzyMatch( fuzzyCDidPtr fid1, fuzzyCDidPtr fid2 );

// SUBROUTINES

void CreateFuzzyId( fuzzyCDidPtr fid, cdPtr cd )

```

FLRO032B.WP

```

{
    long        i;

    // first copy in the number of tracks
    fid->numTracks = cd->numTracks;

5   for(i=0;i<fid->numTracks;i++) {
        // shift left and create a MSB length thats not exact
        fid->fuzzlength[i] = (short)(cd->track[i].lengthMs>>8);
    }
}

10 float FuzzyMatch( fuzzyCDidPtr fid1, fuzzyCDidPtr fid2 )
{
    long        fidmatcherr = 0, fidmatchtotal = 0;
    short       i, trackcnt;
    float       matchpercent;

15     // find the larger number of tracks
    trackcnt = fid1->numTracks < fid2->numTracks ? fid2->numTracks :
fid1->numTracks;

    // cycle thru the tracks accumulating error and total comparedtimes
    for(i=0;i<trackcnt;i++) {
20         if ((i < fid1->numTracks) && (i < fid2->numTracks)) {
            fidmatcherr += abs(fid1->fuzzlength[i] - fid2->fuzzlength[i]);
            fidmatchtotal += fid1->fuzzlength[i];
        } else if (i >= fid2->numTracks) {
            fidmatcherr += fid1->fuzzlength[i];
25         fidmatchtotal += fid1->fuzzlength[i];
        } else if (i >= fid1->numTracks) {
            fidmatcherr += fid2->fuzzlength[i];
            fidmatchtotal += fid2->fuzzlength[i];
        }
    }

30     }

    if (fidmatcherr > 0) {
        matchpercent = 100 - (((float)fidmatcherr/(float)fidmatchtotal)*100);
    } else {
        matchpercent = 100;
35     }
    return matchpercent;
}

```

```

void main(void)
{
    short i;
    float matchpercent;

5    // create global structures for two complete cds with up to 100 tracks
    cd cd2id;
    fuzzyCDid fided2id;
    cd cdFromDB;

10    fuzzyCDid fidedFromDB;

    printf ("Test #1 will compare two CDs that are exactly the same\n\n");

    // put in some test values for the cd track lengths
    // since these are in ms, its basically 60000 = 1 minute

15    cd2id.track[0].lengthMs = 121323;
    cd2id.track[1].lengthMs = 234565;
    cd2id.track[2].lengthMs = 566437;
    cd2id.track[3].lengthMs = 245120;
    cd2id.track[4].lengthMs = 20000;
    cd2id.track[5].lengthMs = 120386;
    cd2id.track[6].lengthMs = 323453;
    cd2id.numTracks = 7;

20    for(i=1;i < cd2id.numTracks;i++) {
        printf ("CD #1: Track = %d length in minutes = %f\n",
                i, (float)cd2id.track[i].lengthMs/60000.0);
    }

25    printf("\n");

    cdFromDB.track[0].lengthMs = 121323;
    cdFromDB.track[1].lengthMs = 234565;
    cdFromDB.track[2].lengthMs = 566437;
    cdFromDB.track[3].lengthMs = 245120;
    cdFromDB.track[4].lengthMs = 20000;
    cdFromDB.track[5].lengthMs = 120386;
    cdFromDB.track[6].lengthMs = 323453;
    cdFromDB.numTracks = 7;

30    for(i=1;i < cdFromDB.numTracks;i++) {
        printf ("CD #2: Track = %d length in minutes = %f\n",
                i, (float)cdFromDB.track[i].lengthMs/60000.0);
    }

```



```
}
```

```
CreateFuzzyId( &fidcd2id, &cd2id );  
CreateFuzzyId( &fidcdFromDB, &cdFromDB );
```

```
matchpercent = FuzzyMatch( &fidcd2id, &fidcdFromDB );  
5 printf ("The cd's matchpercent was computed as=%f", matchpercent);  
printf ("\n");  
printf ("\n");
```

```
printf ("Test #2 will compare two cd that are nearly the same\nexcept they have  
diffent # of tracks \n");
```

```
10 // put in some test values for the cd track lengths  
// since these are in ms, its basically 60000 = 1 minute  
cd2id.track[0].lengthMs = 121323;  
cd2id.track[1].lengthMs = 234565;  
cd2id.track[2].lengthMs = 566437;  
15 cd2id.track[3].lengthMs = 245120;  
cd2id.track[4].lengthMs = 20000;  
cd2id.track[5].lengthMs = 120386;
```

```
cd2id.track[6].lengthMs = 323453;  
cd2id.numTracks = 7;
```

```
for(i=1;i<cd2id.numTracks;i++) {  
    printf ("CD #1: Track = %d length in minutes = %f\n",  
15 i, (float)cd2id.track[i].lengthMs/60000.0 );  
}
```

```
25 printf ("\n");
```

```
cdFromDB.track[0].lengthMs = 121323;  
cdFromDB.track[1].lengthMs = 234565;  
cdFromDB.track[2].lengthMs = 566437;  
cdFromDB.track[3].lengthMs = 245120;  
30 cdFromDB.track[4].lengthMs = 20000;  
cdFromDB.track[5].lengthMs = 120386;  
cdFromDB.numTracks = 6;
```

```
for(i=1;i<cdFromDB.numTracks;i++) {  
    printf ("CD #2: Track = %d length in minutes = %f\n",  
35 i, (float)cdFromDB.track[i].lengthMs/60000.0 );  
}
```

```
CreateFuzzyId( &fidcd2id, &cd2id );
```

```

CreateFuzzyId( &fidcdFromDB, &cdFromDB );
matchpercent = FuzzyMatch( &fidcd2id, &fidcdFromDB );

printf ("The cd's matchpercent was computed as= %f",matchpercent);
printf ("\n");
5 printf ("\n");
printf ("Test #3 will compare two cd that are not the same\n\n");

// put in some test values for the cd track lengths
// since these are in ms, its basically 60000 = 1 minute
cd2id.track[0].lengthMs = 34213;
10 cd2id.track[1].lengthMs = 334565;
cd2id.track[2].lengthMs = 231423;
cd2id.track[3].lengthMs = 134122;
cd2id.track[4].lengthMs = 2342;
15 cd2id.track[5].lengthMs = 3487;
cd2id.track[6].lengthMs = .9976;
cd2id.numTracks = 7;

for(i=1;i<cd2id.numTracks;i++) {
    printf ("CD #1: Track = %d length in minutes = %f\n",
        i, (float)cd2id.track[i].lengthMs/60000.0 );
}
20 printf ("\n");
cdFromDB.track[0].lengthMs = 121323;
cdFromDB.track[1].lengthMs = 234565;
cdFromDB.track[2].lengthMs = 566437;
25 cdFromDB.track[3].lengthMs = 245120;
cdFromDB.track[4].lengthMs = 20000;
cdFromDB.track[5].lengthMs = 120386;
cdFromDB.track[6].lengthMs = 323453;
cdFromDB.numTracks = 6;

30 for(i=1;i<cdFromDB.numTracks;i++) {
    printf ("CD #2: Track = %d length in minutes = %f\n",
        i, (float)cdFromDB.track[i].lengthMs/60000.0 );
}

35 CreateFuzzyId( &fidcd2id, &cd2id);
CreateFuzzyId( &fidcdFromDB, &cdFromDB);

matchpercent = FuzzyMatch( &fidcd2id, &fidcdFromDB );
printf ("The cd's matchpercent was computed as= %f",matchpercent);

```

}

APPENDIX B

```

/*
 * EXACT MATCH CD ID
 * - 1996 ION
5  *
 *
 * by Ty Roberts
 */

#include <stdio.h>
10 #include <stdlib.h>
#include <time.h>

struct cdid{
    long    id[2];
};

15 typedef struct cdid cdid, *cdidPtr;

// structure of a cd track with all times stored in milliseconds

struct cdtrack{
    long    beginMs;    // start time in milliseconds
    long    endMs;      // end time in milliseconds
20    long    lengthMs;  //length in Miliseconds
};

typedef struct cdtrack cdtrack, *cdTrackPtr;

struct cd {
    short    numTracks;
25    cdtrack    track[100];
};

typedef struct cd cd, *cdPtr;

void CreateUniqueId( cdidPtr cid, cdPtr cd );

// SUBROUTINES
30 void CreateUniqueId( cdidPtr cid, cdPtr cd )
{
    long        i, t, n;

    t = 0;

```

```

n = 0;

for(i=0;i<cd->numTracks;i++) {
    // shift left and create a MSB length thats not exact
    t += cd->track[i].lengthMs;
5    n += cd->track[i].beginMs + cd->track[i].endMs;
}
cid->id[0] = t<<10+cd->numTracks;
cid->id[1] = n;
}

10 void main(void)
{
    short i;
    short matchtest;

15    // create global structures for two complete cds with up to 100 tracks
    cd cd2id;
    cdid cd2UID;

    cd cdFromDB;
20    cdid cdFromDBUID;

    printf ("Test #1 will compare two cd that are exactly the same\n\n");

    // put in some test values for the cd track lengths
    // since thes are in ms, its basically 60000 = 1 minute
    cd2id.track[0].beginMs = 0;
25    cd2id.track[1].beginMs = 100001;
    cd2id.track[2].beginMs = 231001;
    cd2id.track[3].beginMs = 345001;
    cd2id.track[4].beginMs = 435001;
    cd2id.track[5].beginMs = 460001;
30    cd2id.track[6].beginMs = 590001;

    cd2id.track[0].endMs = 100000;
    cd2id.track[1].endMs = 231000;
    cd2id.track[2].endMs = 345000;
    cd2id.track[3].endMs = 435000;
35    cd2id.track[4].endMs = 460000;
    cd2id.track[5].endMs = 590000;
    cd2id.track[6].endMs = 690000;
    cd2id.track[0].lengthMs = cd2id.track[0].endMs - cd2id.track[0].beginMs;

```

```

cd2id.track[1].lengthMs = cd2id.track[1].endMs - cd2id.track[1].beginMs;
cd2id.track[2].lengthMs = cd2id.track[2].endMs - cd2id.track[2].beginMs;
cd2id.track[3].lengthMs = cd2id.track[3].endMs - cd2id.track[3].beginMs;
cd2id.track[4].lengthMs = cd2id.track[4].endMs - cd2id.track[4].beginMs;
5 cd2id.track[5].lengthMs = cd2id.track[5].endMs - cd2id.track[5].beginMs;
cd2id.track[6].lengthMs = cd2id.track[6].endMs - cd2id.track[6].beginMs;
cd2id.numTracks = 7;

```

```

for(i=1;i < cd2id.numTracks;i++) {
    printf ("CD #1: Track = %d   length inminutes = %f\n", i,
10 (float)cd2id.track[i].lengthMs/60000.0 );
}
printf ("\n");
cdFromDB.track[0].beginMs = 0;
cdFromDB.track[1].beginMs = 100001;
15 cdFromDB.track[2].beginMs = 231001;
cdFromDB.track[3].beginMs = 345001;
cdFromDB.track[4].beginMs = 435001;
cdFromDB.track[5].beginMs = 460001;
cdFromDB.track[6].beginMs = 590001;
20 cdFromDB.track[0].endMs = 100000;
cdFromDB.track[1].endMs = 231000;
cdFromDB.track[2].endMs = 345000;
cdFromDB.track[3].endMs = 435000;
cdFromDB.track[4].endMs = 460000;
25 cdFromDB.track[5].endMs = 590000;
cdFromDB.track[6].endMs = 690000;
cdFromDB.track[0].lengthMs = cd2id.track[0].endMs - cd2id.track[0].beginMs;
cdFromDB.track[1].lengthMs = cd2id.track[1].endMs - cd2id.track[1].beginMs;
cdFromDB.track[2].lengthMs = cd2id.track[2].endMs - cd2id.track[2].beginMs;
30 cdFromDB.track[3].lengthMs = cd2id.track[3].endMs - cd2id.track[3].beginMs;
cdFromDB.track[4].lengthMs = cd2id.track[4].endMs - cd2id.track[4].beginMs;
cdFromDB.track[5].lengthMs = cd2id.track[5].endMs - cd2id.track[5].beginMs;
cdFromDB.track[6].lengthMs = cd2id.track[6].endMs - cd2id.track[6].beginMs;
cdFromDB.numTracks = 7;

```

```

35
for(i=1;i < cdFromDB.numTracks;i++) {
    printf ("CD #2: Track = %d   length inminutes = %f\n", i,
(float)cdFromDB.track[i].lengthMs/60000.0 );
}

```

```

40 CreateUniqueId( &cd2UID, &cd2id );

```

```

printf( "Unique ID for CD #1 = %d%d\n", cd2UID.id[0], cd2UID.id[1] );

CreateUniqueId( &cdFromDBUID, &cdFromDB );
printf( "Unique ID for CD #2 = %d%d\n", cdFromDBUID.id[0],
cdFromDBUID.id[1] );

5      matchtest = (cd2UID.id[0] == cdFromDBUID.id[0]) && (cd2UID.id[1] ==
cdFromDBUID.id[1]);

printf( "The cd's match if result is non zero matchresult=%d", matchtest);

printf( "\n");

printf( "\n");
10      printf( "Test #2 will compare two cd that are nearly the same\nexcept they have
diffent # of tracks \n");

// put in some test values for the cd track lengths
// since thes are in ms, its basically 60000 = 1 minute
cd2id.track[0].beginMs = 0;
15      cd2id.track[1].beginMs = 100001;
cd2id.track[2].beginMs = 231001;
cd2id.track[3].beginMs = 345001;
cd2id.track[4].beginMs = 435001;
cd2id.track[5].beginMs = 460001;
20      cd2id.track[6].beginMs = 590001;
cd2id.track[0].endMs = 100000;
cd2id.track[1].endMs = 231000;
cd2id.track[2].endMs = 345000;
cd2id.track[3].endMs = 435000;
25      cd2id.track[4].endMs = 460000;
cd2id.track[5].endMs = 590000;
cd2id.track[6].endMs = 690000;

cd2id.track[0].lengthMs = cd2id.track[0].endMs - cd2id.track[0].beginMs;
cd2id.track[1].lengthMs = cd2id.track[1].endMs - cd2id.track[1].beginMs;
30      cd2id.track[2].lengthMs = cd2id.track[2].endMs - cd2id.track[2].beginMs;
cd2id.track[3].lengthMs = cd2id.track[3].endMs - cd2id.track[3].beginMs;
cd2id.track[4].lengthMs = cd2id.track[4].endMs - cd2id.track[4].beginMs;
cd2id.track[5].lengthMs = cd2id.track[5].endMs - cd2id.track[5].beginMs;
cd2id.track[6].lengthMs = cd2id.track[6].endMs - cd2id.track[6].beginMs;
35      cd2id.numTracks = 7;

for(i=1;i<cd2id.numTracks;i++) {

```

```

        printf ("CD #1: Track = %d  length inminutes = %f\n", i,
(float)cd2id.track[i].lengthMs/60000.0 );
    }

```

```

    printf ("\n");
5   cdFromDB.track[0].beginMs = 0;
    cdFromDB.track[1].beginMs = 100001;
    cdFromDB.track[2].beginMs = 231001;
    cdFromDB.track[3].beginMs = 345001;
    cdFromDB.track[4].beginMs = 435001;
10  cdFromDB.track[5].beginMs = 460001;
    cdFromDB.track[6].beginMs = 590001;

```

```

    cdFromDB.track[0].endMs = 100000;
    cdFromDB.track[1].endMs = 231000;
15  cdFromDB.track[2].endMs = 345000;
    cdFromDB.track[3].endMs = 435000;
    cdFromDB.track[4].endMs = 460000;
    cdFromDB.track[5].endMs = 590000;

```

```

20  cdFromDB.track[0].lengthMs = cd2id.track[0].endMs - cd2id.track[0].beginMs;
    cdFromDB.track[1].lengthMs = cd2id.track[1].endMs - cd2id.track[1].beginMs;
    cdFromDB.track[2].lengthMs = cd2id.track[2].endMs - cd2id.track[2].beginMs;
    cdFromDB.track[3].lengthMs = cd2id.track[3].endMs - cd2id.track[3].beginMs;
    cdFromDB.track[4].lengthMs = cd2id.track[4].endMs - cd2id.track[4].beginMs;
25  cdFromDB.track[5].lengthMs = cd2id.track[5].endMs - cd2id.track[5].beginMs;
    cdFromDB.numTracks = 6;

```

```

    for(i=1;i < cdFromDB.numTracks;i++) {
        printf ("CD #2: Track = %d  length inminutes = %f\n", i,
30  (float)cdFromDB.track[i].lengthMs/60000.0 );
    }

```

```

    CreateUniqueId( &cd2UID, &cd2id );
    printf( "Unique ID for CD #1 = %d%d\n", cd2UID.id[0], cd2UID.id[1] );
35

```

```

    CreateUniqueId( &cdFromDBUID, &cdFromDB );
    printf( "Unique ID for CD #2 = %d%d\n", cdFromDBUID.id[0],
40  cdFromDBUID.id[1] );

```

```

    matchtest = (cd2UID.id[0] == cdFromDBUID.id[0]) && (cd2UID.id[1] ==
cdFromDBUID.id[1]);

```

```

    printf ("The cd's match if result is non zero matchresult=%d",matchtest);

```


}

	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---

CLAIMS

I claim:

1. A method for synchronizing the delivery of visual content from a first computer to a second computer with the playing of a musical recording on said second computer, said computers being connected by means of a computer network, said method comprising the steps of:

5 activating by means of information transferred over said network from said first
 computer a computer program on said second computer;
by means of said computer program, reporting over said network to said first
 computer a position indicator identifying a position on a musical recording
 currently being played on said second computer, said position indicator
10 including an identification of a track on said musical recording and a time
 within said track;
delivering a unit of visual content to said second computer over a network
 connection, said unit of visual content corresponding to one or more
 frames of visual information and corresponding to said position
15 information;
receiving said unit of visual content by said second computer;
following receipt of said unit of visual content, determining a second position
 indicator identifying a second position on said musical recording; and
displaying said unit of visual content if said unit of visual content corresponds
20 approximately to said second position indicator.

ABSTRACT

Entertainment content complementary to a musical recording is delivered to a user's computer by means of a computer network link. The user employs a browser to access the computer network. A plug-in for the browser is able to control an audio CD or other device for playing the musical recording. A script stored on the remote computer accessed over the network is downloaded. The script synchronizes the delivery of the complementary entertainment content with the play of the musical recording.

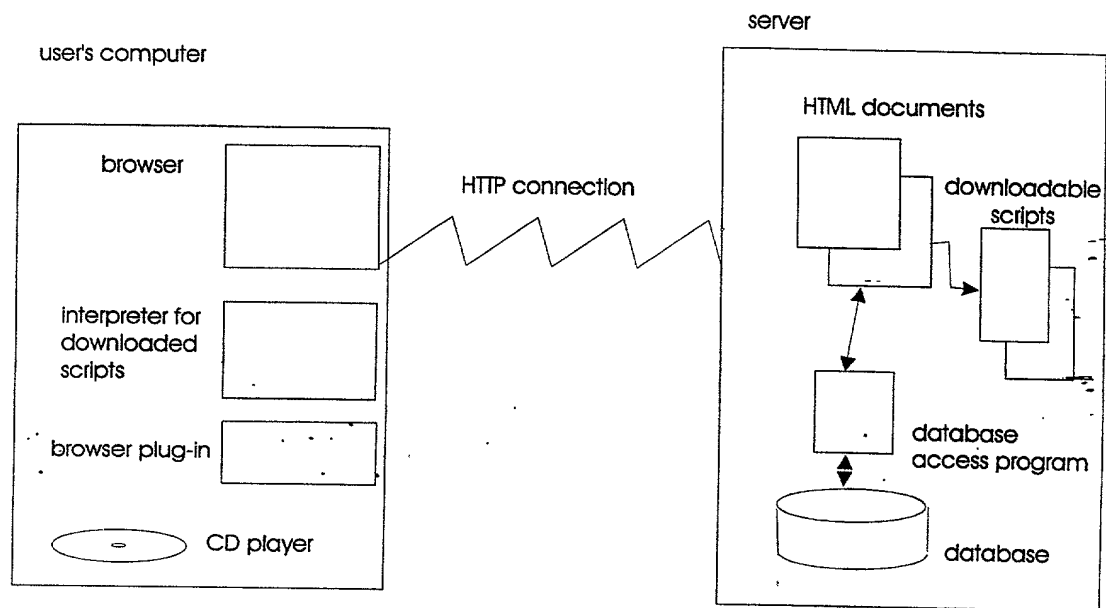


Figure 1

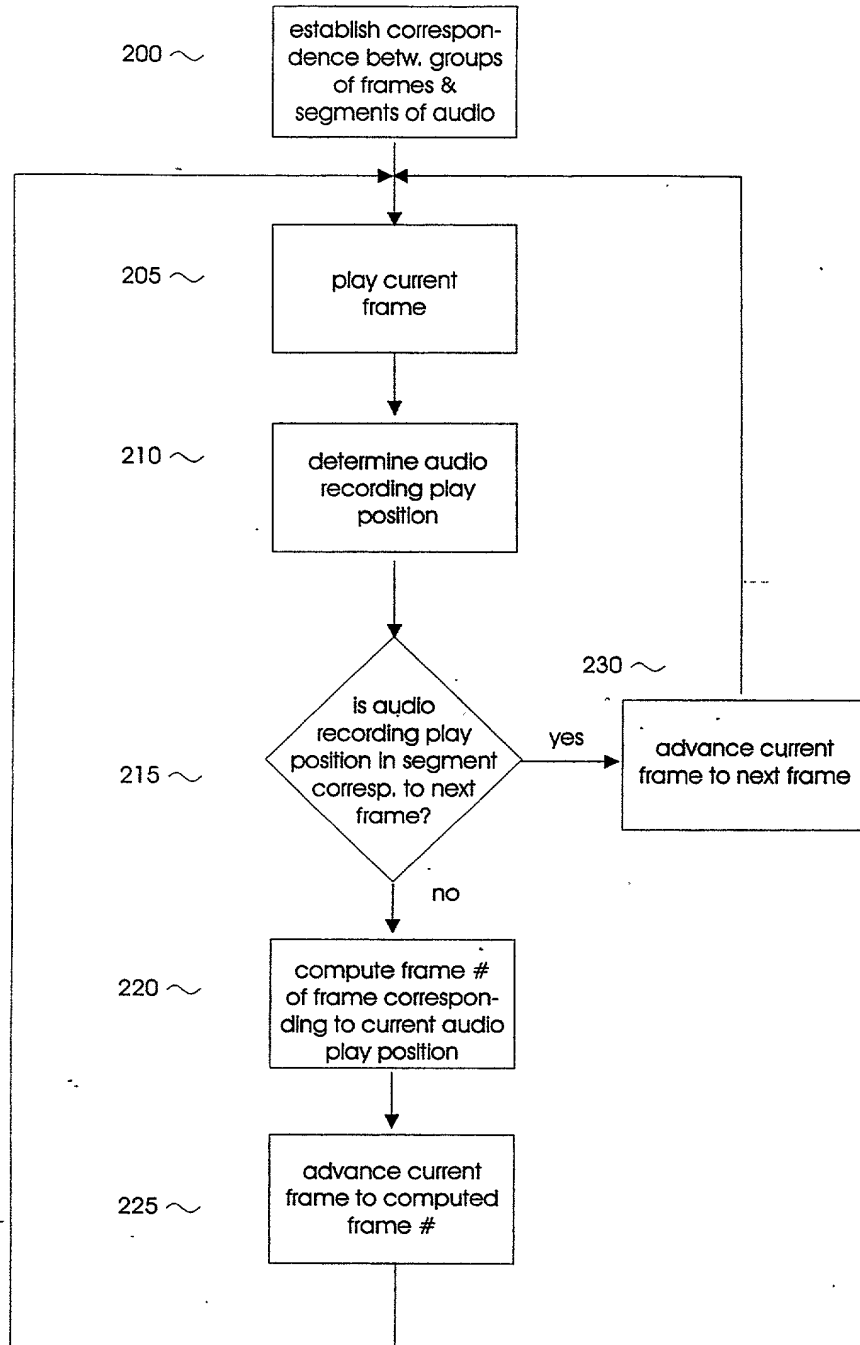


Figure 2

63960" 63960" 63960"

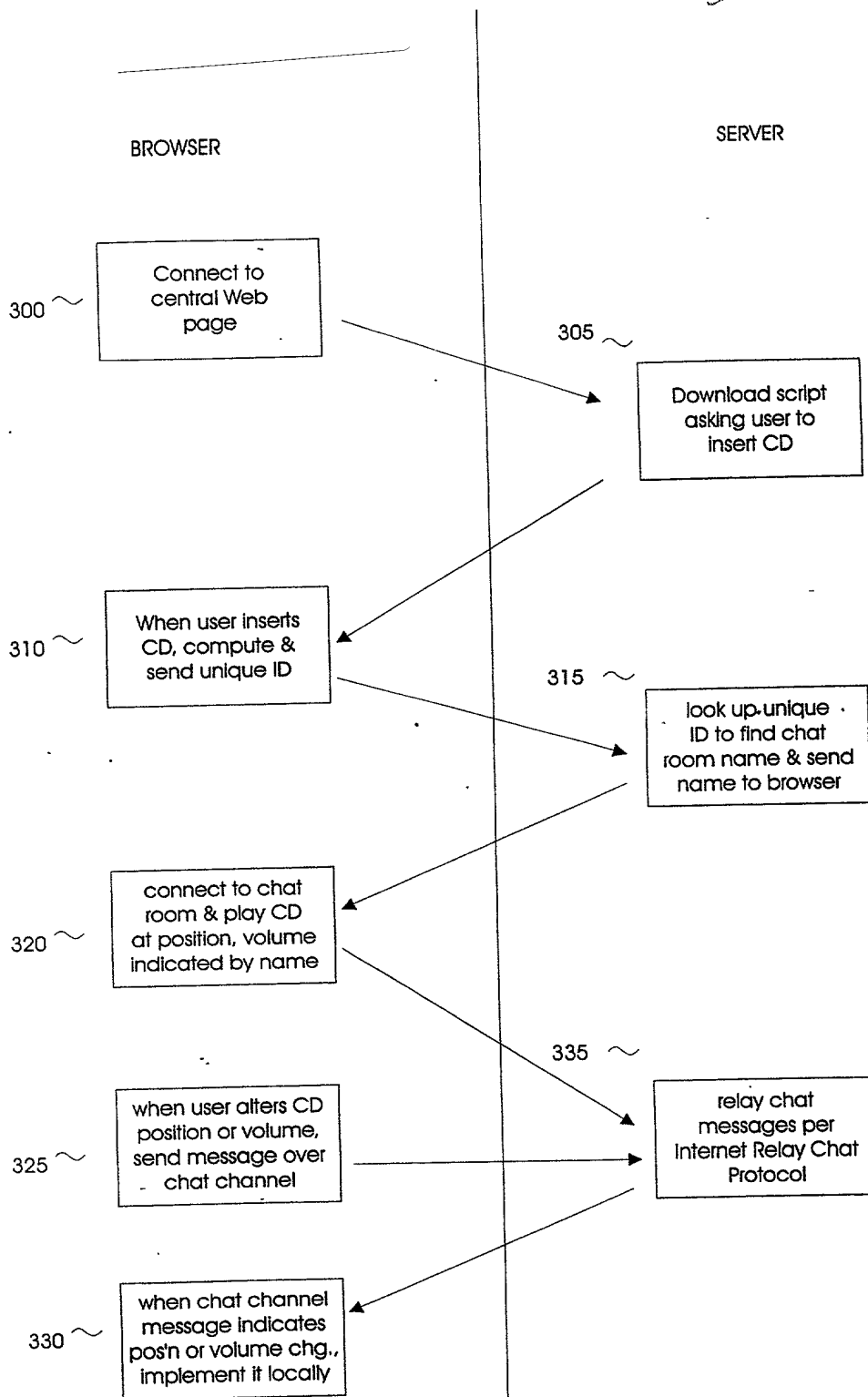
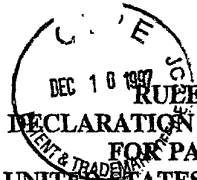


Figure 3



DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

As a below named inventor, I hereby declare that my residence, post office address and citizenship are as stated below next to my name, and I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled NETWORK DELIVERY OF INTERACTIVE ENTERTAINMENT COMPLEMENTING AUDIO RECORDINGS the specification of which (CHECK applicable BOX(ES)).

_____ is attached hereto.
X was filed on 4/15/97 as U.S. application Serial No. 08/838,082
_____ was filed as PCT international application No. PCT/_____/_____ on _____ and (if applicable to U.S. or PCT application) was amended on _____.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to the examination of this application in accordance with 37 C.F.R. 1.56(a) and 35 U.S.C. 102. I hereby claim foreign priority benefits under 35 U.S.C. 119/365 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date (1) before that of the application on which priority is claimed, or (2) if no priority claimed, before the filing date of this application:

PRIOR FOREIGN APPLICATION(S)

PRIORITY CLAIMED
YES NO

Number Country Day/MONTH/Year Filed

I hereby claim the benefit under 35 U.S.C. 120/365 of all United States and PCT international applications listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in such prior applications in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose material information as defined in 37 C.F.R. 1.56(a) which occurred between the filing date of the prior applications and the national or PCT international filing date of this application:

PRIOR U.S. OR PCT APPLICATION(S)

Application Serial No. Day/MONTH/Year Status: patented, pending, abandoned

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

And I hereby appoint IRELL & MANELLA LLP, 1800 Avenue of the Stars, Suite 900, Los Angeles, California 90067, telephone number (310) 277-1010 (to whom all communications are to be directed) and Norman E. Brunell, Registration No. 26,533; Robert Steinberg, Registration No. 33,144; Bruce D. Kuyper, Registration No. 33,937; Gary Frischling, Registration No. 35,515; Soyeon Pak Laub, Registration No. 39,266; Flavio Rose, Registration No. 40,791; and Sam Lu, Registration No. 40,707; (all of the same address), and any and all attorneys associated therewith after this date, individually and collectively my attorneys to transact all business in the Patent and Trademark Office connected therewith and with the resulting patent.

1) INVENTOR'S SIGNATURE Dale Tyson Roberts Date December 8th, 1997
Inventor's Name (typed) DALE TYSON ROBERTS United States
First MI Family Name Citizenship

Residence Sausalito (State/Foreign Country) California 94965
Post Office Address (Include Zip Code) 164 Spencer Avenue

2) INVENTOR'S SIGNATURE Ann E. Greenberg Date Dec 8, 1997
Inventor's Name (typed) ANN E. GREENBERG United States
First MI Family Name Citizenship

Residence San Anselmo (State/Foreign Country) California 94560
Post Office Address (Include Zip Code) 216 San Francisco Boulevard

3) INVENTOR'S SIGNATURE _____ Date _____
Inventor's Name (typed) _____
First MI Family Name Citizenship

Residence _____ (State/Foreign Country) _____
Post Office Address (Include Zip Code) _____